**lezione 19/2/2024**

***programma***: txt scritto con sintassi (c, java…) che vanno compilati(procedura di caricamento, traduzione ed esecuzione…)

***script:*** testo con sintassi ma non va compilato (a volte sono usati in caso di operazioni RAD, questione di velocità, possono essere multipiattaforma: es Python)

cio si puo fare su linux server(nella shell)

**—------------------------------------------------------------------------------------------------------------------------**

**Ripasso** (check chatGPT e https://linuxcommand.org)

<https://linuxcommand.org/>

comandi:

ls: Elenco dei file e delle directory nel percorso specificato.

* cd: Cambia la directory.
* pwd: Mostra la directory corrente.
* mkdir: Crea una nuova directory.
* rm: Rimuove file o directory.
* cp: Copia file e directory.
* mv: Sposta o rinomina file e directory.
* cat: Visualizza il contenuto di un file.
* nano o vi/vim: Editor di testo per modificare i file.
* grep: Cerca pattern nei file.
* chmod: Modifica i permessi dei file.
* chown: Cambia il proprietario dei file.
* sudo: Esegui un comando come superutente (root).
* man: manuale del sistema operativo

directory:

* **/bin:** Contiene i programmi di base del sistema.
* /**boot**: Contiene i file necessari per l'avvio del sistema, come il kernel e i file di configurazione del bootloader GRUB.
* /**dev**: Contiene i file che rappresentano i dispositivi hardware del sistema.
* /**etc**: Contiene i file di configurazione del sistema e dei programmi.
* /home: La cartella principale per gli utenti del sistema.
* /**lib** e /**lib64**: Contengono le librerie condivise utilizzate dai programmi del sistema.
* /**media**: Cartella utilizzata per montare dispositivi di archiviazione rimovibili.
* /**mnt**: Cartella utilizzata per montare temporaneamente dispositivi di archiviazione.
* /**opt**: Utilizzato per installazioni di software aggiuntivo di terze parti.
* /**proc**: Contiene informazioni sulle attività del sistema.
* /**root**: La cartella home dell'utente root.
* /**run**: Contiene file di sistema temporanei generati durante l'avvio.
* /**sbin**: Contiene programmi di sistema. (ci sono comandi che vanno eseguiti con super user e servono per amministrazione di sistema)
* /**srv**: Contiene dati dei servizi forniti da questo sistema.
* /**sys**: Contiene file di sistema virtuali utilizzati per interagire con il kernel.
* /**tmp**: Contiene file temporanei utilizzati da programmi e processi temporanei.
* /**usr**: Contiene i programmi, le librerie e la documentazione non essenziali per il sistema.
* /**var**: Contiene file di dati vari, come file di log, e-mail, database, ecc.

le distribuzioni di linux hanno in comune il **kernel**;  
linux ha un **kernel monolitico e modulare** (kernel: intermediario tra OS e HW);  
ogni cosa in linux server è un file;

repository: contenitori di artefatti(prog compilato/ cod sorgente/ libreria/ …) sono usati per fare il versionamento(avere a disp le versioni del programma) (es github)

driver: programma legato ai dispositivi e al loro funzionamento;

----------------------------------------------------------------------------------------------------------------------------------------------------

pipeline: concatena più comandi inseriti in una riga |

history: cronologia comandi

swap: (cambiare) area che la macchina crea nella memoria fisica, dove scarica roba temporanea se non riesce a mettere nella ram(RAM[piena] CPU → Memoria/disco[swap])  
 xché serviva? per ram insufficiente.

lettura dei oggetti::

‘d’: directory / ‘-’ : file / ‘x’: (nella directory) accesso libero a tuttu

dopo ‘ r, w, x’ : Sono i permessi per usare l’oggetto (Read, Write; eXecutable)

la visualizzazione dei permessi è divisa in proprietario / gruppo / altri utenti   
al posto di scrivere 9 volte le lettere dei diritti si usano 3 serie di numeri(che vengono convertiti in binario)  
 777, 555, 000,100, 702……

comando per modificare i permessi: chmod  
comando per cambiare proprietario: chown

comando per cambiare gruppo: chgrp

se volessimo connetterci virtualmente a un terminale bisogna:

1-verificare protocollo da usare: SSH (lvl ⅞ della pila ISO/OSI)   
 (usato ‘service ssh status’ per controllare stato ssh o altre funzioni grazie alla voce  
 service);

2- service ssh: da tutte le funzioni concesse; (NB ogni funzione ha un ID per facilitare ricerca)

3-in vb, porte, gestore NAT, porta host/guest 22;

4-terminale Windows(ipconfig, copiare vehternet NAT/ accesso usando ssh nomehost@172.24.0.1)

5- avviare mobaxterm, sessione, configura tutto

6-terminale della vb collegato a un terminale “remoto”(quello di mobaxterm)

NAT(Network addres translation): mezzo per tradurre reti, tabella che va a cambiare l'indirizzo ip che viene usato per connettersi da qualche parte

variabili d’ambiente:variabili relative all’ambiente di lavoro corrente

comandi:

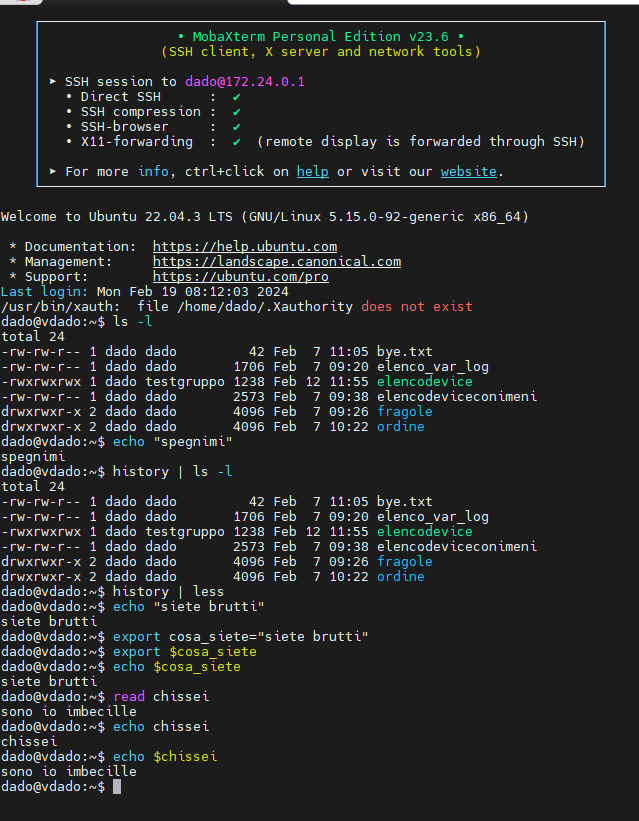
echo “” :scrive output;

export variabile=”contenuto”

echo $variabile

read variabile -> accapo inserire roba

echo $variabile :da roba inserita



nei file bash si commenta con #;

con

#!/bin/bash

si esegue senza sh un programmino che scriviamo (adesso in vs code)

chmod +permesso

check script:

#!/bin/bash

# - chiedere percorso che vogliamo backuppare

# - crea directory per backup nella home (check se esiste già)

# - nome file backup deve avere data e ora dell'operazione (es: "backup-YYYYMMDD-HHMMSS.tar.gz")

# - deve dare conferma dell'operazione

# 1 chiederw cosa backuppare

echo "su quale directory devo eseguire il backup?"

read directory

#2 crea directory per backup/check se esiste già

#controlliamo se la dir esiste

if [ ! -d "$directory" ]; then

echo "la directory non esiste"

exit 1

fi

# gestiamo dir di backup in home con i controlli del caso

#creiamo la variabile nella dir di backup

backup\_dir="$HOME/Backup"

#controllo se esiste già altrimenti la creo

if [ ! -d "$backup\_dir" ]; then

echo "la directory non esisteva fino ad adesso :)"

mkdir -p "$backup\_dir"

fi

#3 formattiamo data e ora corrente per il nome del backup

date\_time\_now=$(date "+%Y%m%d-%H%M%S")

#montiamo nome file di backup

file\_backup="backup-$date\_time\_now.tar.gz"

#creo il file backup

tar -czf "$backup\_dir/$file\_backup" "$directory"

#4 creiamo la conferma

echo "backup eseguito della '$directory' dentro '$backup\_dir/$file\_backup' confermi operazione? "